FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 30, No 4 (2015), 401–417

# AN ENHANCED FIREFLY ALGORITHM FOR MIXED VARIABLE STRUCTURAL OPTIMIZATION PROBLEMS *

**Ivona Brajevic and Jelena Ignjatović**

**Abstract.** The recently developed firefly algorithm has become one of the prominent population-based metaheuristics due to its efficiency in solving a wide range of diverse real-world problems. In this paper an enhanced firefly algorithm to solve mixed variable structural optimization problems is presented. Two modifications related to the constraint handling method based on Deb's rules and the geometric progression reduction scheme mechanism are introduced in order to improve its performance in the constrained search space. The proposed algorithm is tested on four classical structural optimization problems taken from literature. The obtained results show that the proposed approach was very competitive in the considered problems, mostly outperforming the original firefly algorithm.

## 1. Introduction

Most structural optimization problems have nonlinear objective functions and nonlinear constraints. The solution complexity is additionally increased since these problems often include continuous variables and discrete variables. Generally, structural optimization problems can be considered as constrained optimization problems.

A general constrained optimization problem is to find $x$ so as to:

$$\text{minimize } f(x),\ x = (x_1, x_2, \dots, x_D) \in R^D \tag{1.1}$$

subject to:

$$\begin{aligned} g_j(x) &\le 0, \quad j = 1, \dots, q \\ h_j(x) &= 0, \quad j = q+1, \dots, m \end{aligned} \tag{1.2}$$

Received April 09, 2015; Accepted June 16, 2015
2010 *Mathematics Subject Classification.* Primary 90C30
*This research is supported by Ministry of Education and Science, Republic of Serbia, Project No. III-44006

where $q$ is the number of inequality constraints and $m - q$ is the number of equality constraints for a given problem. Each parameter $x_i$, $i = 1, 2, ..., D$ is limited by its lower and upper bounds $l_i \leq x_i \leq u_i$ which define the search space $S \subseteq R^D$. A solution is feasible if it satisfies all constraints, while an infeasible solution does not satisfy at least one constraint.

Solving mixed variable structural optimization problems usually requires global search techniques that are problem-specific [8]. The main lacks of the classical optimization algorithms are their inefficiency in solving highly non-linear problems and their inflexibility to adapt the solution algorithm to a given problem [2]. On the other hand metaheuristic optimization algorithms can be successfully applied to solve these types of problems.

Metaheuristic techniques are global optimization methods which can be modified to suit specific problem requirements [20]. Population-based metaheuristics are the important class of metaheuristics which find good solutions by iteratively selecting and then combining existing solutions from a set, usually called the population. In these metaheuristic algorithms exploration and exploitation represent the two cornerstones of problem solving [19]. Exploration refers to visiting new regions of the search space, while exploitation refers to moves that focus searching those regions of a search space within the neighborhood of previously visited solutions. Strong exploration increases the probability of discovering the global optimum, while too much exploitation tends to make the algorithm being trapped in a local optimum [23]. For a search algorithm good ratio between exploration and exploitation is essentially important in order to achieve good optimization performance.

The most prominent members of this class are evolutionary algorithms and swarm intelligence algorithms. The evolutionary algorithms employ iterative progress of a population of solutions based on some mechanisms inspired by biological evolution, such as reproduction, mutation, recombination and selection. Examples of the most popular evolutionary algorithms are genetic algorithms (GA) [11], differential evolution (DE) [17] and evolution strategies (ES) [3]. Swarm intelligence techniques are based on mimicking the so-called swarm intelligence characteristics of biological agents such as birds, fish, humans and others. Some of the most prominent swarm intelligence optimization methods include particle swarm optimization (PSO) [13], artificial bee colony (ABC) [12, 4], firefly algorithm (FA) [22, 8], cuckoo search (CS) [25, 10] and bat algorithm (BA) [24, 9].

The FA is one of the recent swarm intelligence algorithms inspired by the flashing behavior of fireflies which has been shown as very efficient in dealing with global optimization problems. The FA results for unconstrained optimization problems indicated that it was superior to PSO and GA in terms of efficiency and success rate [22]. As the most basic variants of the metaheuristics, the original FA also lacks a mechanism to deal with the constraints of a numerical optimization problem. In [8] it was extended to solve structural optimization problems by adding a constraint handling technique based on penalty approach. The optimization results showed that FA is more efficient than other metaheuristic approaches such as PSO, GA, DE and simulated annealing (SA).

In the last five years the number of researchers being interested in the FA algorithm have increased rapidly [23]. Although the FA was originally proposed for solving numerical problems, the modified versions have been also described for the discrete and combinatorial types of problems. Nowadays, the FA and its variants has been used in many applications in several different areas, such as image processing [5, 28], industrial optimization [26], antenna design [27], civil engineering [18], robotics [16] etc.

In this paper an enhanced firefly algorithm (E-FA) is proposed with the goal to improve the performance of the FA in solving mixed variable structural optimization problems. In order to reduce diversity in the population faster and hence increase the exploitation ability of the FA, the proposed approach employs Deb's rules as constraint handling method and introduces geometric progression reduction scheme similar to the cooling schedule of simulated annealing. The E-FA is tested on the four mixed variable structural optimization problems and the obtained results are compared to the same of the FA and to those from several state-of-the-art algorithms.

The rest of the paper is organized as follows. The detailed description of the firefly algorithm is in section "The firefly algorithm". The proposed enhanced firefly algorithm is described in section "The proposed approach: E-FA". Section "Benchmark problems" presents the four structural optimization problems. The parameter settings and analysis of the obtained results are presented in section "Experimental analysis".

## 2. The firefly algorithm

The FA formulated by Yang [22] is a swarm intelligence algorithm inspired by the flashing behavior of fireflies. Fireflies are characterized by their flashing light. The production of these lights is performed by a complicated set of chemical reactions. Their fundamental functions are attracting mating partners or for protection against predators. The intensity of a firefly lights decrease when the distance from the light source increases [7]. Also, air absorbs the light as the distance from the source increases. This firefly behavior is modeled into the FA so that the light intensity is proportional to the objective function of the problem to be optimized.

Considering the fact that an adaptation of the natural behavior of the fireflies in an algorithm is very complex, the next idealized rules are assumed by constructing of the FA [7]:

- All fireflies are unisex.
- Their attractiveness is proportional to their brightness.
- The brightness of a firefly is associated with the objective function.

The main steps of the FA [21] are presented as follows.

**Step 1.** (Generate the initial population of solutions)
The FA generates randomly initial population of solutions, $x_{ik}$, $i = 1, 2, \ldots, SP$, $k = 1, 2, \ldots, D$:

$$x_{ik} = l_k + \mathrm{rand}(0, 1) \cdot (u_k - l_k) \tag{2.1}$$

where $SP$ is the size of population, $D$ is dimension of the problem, $k \in 1, 2, \ldots, D$, $l_k$ and $u_k$ are the lower and upper bound of the parameter $x_{ik}$ and $rand(0, 1)$ is an uniformly distributed random number between 0 and 1. After the generation of initial population, the objective function values for all solutions $x_i$ are calculated and variable $t$ is set to 1.

**Step 2.** (Calculate the new population)
Each solution of the new population is created from the appropriate solution $x_i$ in the following way:

For each solution $x_i$, algorithm examines every solution $x_j$, $j = 1, 2, ..., i$, iteratively, starting from $j = 1$. If solution $x_j$ has higher objective function value than $x_i$ ($x_j$ is brighter than $x_i$), the parameter values $x_{ik}$, $k = 1, 2, ..., D$, are updated by:

$$x_{ik} = x_{ik} + \beta \cdot (x_{ik} - x_{jk}) + \alpha \cdot S_k \cdot (rand_k - \frac{1}{2}) \tag{2.2}$$

where the second term is due to the attraction and the third term is randomization term.

In the second term of Eq. 2.2, parameter $\beta$ is the attractiveness of fireflies and in [21] the selected monotonically decreasing function which describes firefly attractiveness is the exponential function:

$$\beta = \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \tag{2.3}$$

where $r_{ij}$ is the distance between firefly $i$ and firefly $j$, while $\beta_0$ and $\gamma$ are predetermined algorithm parameters: maximum attractiveness value and absorption coefficient, respectively.
Distance $r_{ij}$ between fireflies $i$ and $j$ is obtained by Cartesian distance by:

$$r_{ij} = \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2} \tag{2.4}$$

Control parameter $\beta_0$ describes attractiveness when two fireflies are found at the same point of search space, i.e. at $r = 0$. The variation of attractiveness with increasing distance from a communicated firefly is determined by the control parameter $\gamma$. The value of parameter $\gamma$ is essentially important in determining the speed of the convergence and how the FA algorithm behaves.

In the third term of Eq. 2.2, $\alpha \in [0, 1]$ is randomization parameter, $S_k$ are the scaling parameters and $rand_k$ is a random number uniformly distributed between 0 and 1. The scaling parameters $S_k$ are calculated by:

$$S_k = |u_k - l_k| \tag{2.5}$$

In addition, whenever the values of the solution $x_i$ are changed, the FA controls the boundary conditions of the created solution and memorizes the new objective function value instead of the old one. The boundary constraint-handling mechanism used in the FA is given by:

$$x_{ik} = \begin{cases} l_k & \text{, if } x_{ik} < l_k \\ u_k & \text{, if } x_{ik} > u_k \end{cases} \tag{2.6}$$

Last solution obtained by Eq. 2.2 is the final solution of the new population which will be transfered in the next iteration of the FA.

**Step 3.** (Reduce the randomization parameter)

In [21] it was found that the solution quality can be enhanced by reducing the randomization parameter $\alpha$ with a geometric progression reduction scheme which can be described by:

$$\alpha(t) = \alpha(t-1) \cdot \theta^{\frac{1}{MCN}} \tag{2.7}$$

where $MCN$ is maximum cycle number, $t$ is the current iteration number, $\theta$ is the parameter calculated by:

$$\theta = \frac{10.0^{-4.0}}{0.9} \tag{2.8}$$

However, this step is optional in the FA.

**Step 4.** (Record the best solution)

Rank the fireflies by their light intensity/objectives and memorize the best solution so far $x_{best}$ and increase the variable $t$ by one.

**Step 5.** (Check the termination criterion)

If the $t$ is equal to the maximum number of iterations then finish the algorithm, else go to Step 2.

In the FA there are three parameters that control its behavior i.e., the step size of randomized move $\alpha$, the attractiveness $\beta$ and the absorption coefficient $\gamma$. The remaining control parameters are the size of population $SP$ and the maximum cycle number $MCN$ which are common for all population based metaheuristics.

The control parameter values of the FA may affect convergence behavior at different extents [23]. The control parameter $\alpha$ control the randomness or to some extent the diversity of solutions. It was found that for most applications the performances of the FA can be improved by using a geometric progression reduction scheme similar to the cooling schedule of simulated annealing which is described by Eq.2.7. The control parameter $\beta$ controls the attractiveness, and for most applications it was found that $\beta_0 = 1$ can be used. The control parameter $\gamma$ in the majority of applications typically varies from 0.01 to 100. This parameter can be related to the scaling $L$, where $L$ is the average scale of the problem of interest. If the scaling variations are significant, than it is usually set $\gamma = 1/\sqrt{L}$, otherwise the common setting for this parameter is $\gamma = O(1)$. The best range of population size is from 25 to 40. The values higher than 50 are not recommended since it will significantly increase computation time [8].

## 3. The proposed algorithm: E-FA

In the proposed E-FA two modifications are introduced in comparison to the original FA. Firstly, the three feasibility based rules are employed in order to guide the search to the feasible region of the search space. The second modification is using the geometric progression reduction scheme to reduce the scaling factors $S_k$ gradually. The details of each modification are explained in the further text.

### 3.1. The modification related to Deb's rules

The original variant of the FA was not designed to deal with constrained search space. Therefore in order to solve constrained optimization problems the penalty function approach was incorporated in the FA [8]. The use of penalty functions is the most common approach employed to deal with constrained search spaces because a constrained problem is solved as an unconstrained one. However, the main drawback of this approach is that it requires a careful fine tuning of the penalty factors that estimate the degree of penalization to be applied [15]. The lack of penalty approach was one of the reasons of using the constraint-handling method based on three feasibility rules in the E-FA.

The set of three feasibility rules [6], also called Deb's rules, where two solutions are compared at a time defined by Deb are:

- Any feasible solution satisfying all constraints is preferred to any infeasible solution violating any of the constraints.
- Among two feasible solutions, the one having better fitness value is preferred.

- If both solutions are infeasible, the one with the lower sum of constraint violation is preferred, where the sum of constraint violation is defined as:

$$CV(x) = \sum_{j=1}^{q} max(0, g_j(x)) + \sum_{j=q+1}^{m} |h_j(x)| \tag{3.1}$$

where $g_j(x)$ are the inequality constraints, $h_j(x)$ are the equality constraints, $q$ is the number of inequality constraints and $m$ is the total number of inequality.

In the proposed E-FA the selection mechanism based on Deb's rules is used two times during the creation a new solution of the population. Firstly, these rules are employed instead of the greedy selection in order to decide which firefly is brighter. Secondly, Deb's rules are used each time after the Eq.2.4 is applied in order to decide whether the solution will be updated.

In [12] it was noticed that the algorithms which use Deb's rules lack diversity in the population because feasible solutions are preferred to infeasible ones. On the other hand, a selection mechanism is one of the major components which have direct influence on the performances of a search algorithm [19]. Hence, in the E-FA the diversity of the population is significantly decreased in comparison with the original FA considering additional usage of Deb's rules in order to decide whether the solution will be updated.

From the exploration and exploitation viewpoints, decreasing diversity of the population during the search process increases the exploitation ability of a search algorithm [19]. Also, empirical knowledge from observations of the convergence behavior of common optimization algorithms suggests that exploitation tends to increase the speed of convergence [23]. Hence, the modification of the FA related to Deb's rules increases exploitation and consequently convergence speed of the FA.

### 3.2. The modification related to scaling factors

In the original firefly algorithm it was found that it is possible to improve the solution quality by reducing the randomness gradually [22]. It was also concluded that a further improvement on the convergence of the algorithm is to vary the randomization parameter so that it decreases gradually as the optimum is being reached. Therefore, in the FA used to solve constrained optimization problems reducing the randomization parameter by the geometric progression reduction scheme is proposed [21].

Inspired by these conclusions in the E-FA it was found that the quality of the results and convergence speed can be further improved by reducing each scaling parameter $S_k$, $k = 1, 2, ..., D$, by using the same geometric progression reduction scheme which is used to decrease the parameter $\alpha$, i.e. the scheme can be described by:

$$S_k(t) = S_k(t-1) \cdot \theta^{\frac{1}{MCN}} \tag{3.2}$$

where $MCN$ is maximum cycle number, $t$ is the current iteration number, $\theta$ is the parameter calculated by Eq. 2.8.

### 3.3. The pseudo-code of the E-FA

The pseudo code of the E-FA is presented as Alg. 1.

It is important to note that both FA and E-FA use the same number of control parameters, since tuning the control parameters of an algorithm might be more difficult than the problem itself [19]. Also, in order to solve the mixed variable structural optimization problems, continuous values of discrete variables were rounded to the nearest available discrete values after evolution according to Eq.2.2, as well as after the initialization phase of the algorithm.

**Algorithm 1** The pseudo code of the E-FA

```
Initialize the population solutions by Eq.2.1
Evaluate each x_i, i = 1, 2, ..., SP
Initialize control parameters SP, MCN, α_0, γ and β_0
Calculate each S_k, k = 1, 2, ..., D by Eq.2.5
t = 1
repeat
  for i = 1 to SP do
    for j = 1 to SP do
      if (x_j is chosen according to Deb's rules when we compare x_i and x_j) then
        for k = 1 to D do
          g_k = x_ik + β · (x_ik − x_jk) + α · S_k · (rand_k − 1/2) {where β is calculated by
          Eq.2.3}
        end for
        if (g is chosen according to Deb's rules when we compare x_i and g) then
          x[i] = g {the new solution g is accepted in the population}
        end if
      end if
    end for
  end for
  for k = 1 to D do
    Calculate new value of S_k by Eq.3.2
  end for
  Calculate new value of α by Eq.2.7
  Memorize the best solution achieved so far
  t = t + 1
until t = MCN
```

## 4. Benchmark problems

The proposed E-FA was applied to four mixed variable structural optimization design problems: welded beam, reinforced concrete beam, helical spring design and stepped cantilever beam, same as used by [8].

**The welded beam design problem**, shown in Figure 4.1, aims to minimize the cost of beam subject to constraints on shear stress, bending stress in the beam, buckling load on the bar and deflection of the beam. The four continuous design variables are the weld thickness $h$, length of the weld $l$, width of the bean $t$ and the thickness of the beam $b$, where $0.1 \leq h, t \leq 2.0, \quad 0.1 \leq l, b \leq 10.0$. The optimum solution is located on the boundaries of the feasible region which is very small with respect to the entire search space. The best reported rounded value-to-reach for this problem is 1.724852 [14].

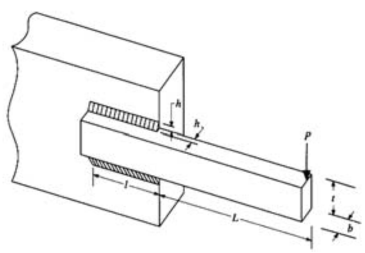

Fig. 4.1: Welded beam design structure

**The reinforced concrete beam design problem** aims to be designed for minimum cost. The beam is supported at two points spaced by 30 ft and it is subject to a live load of 2000 lbf and a dead load of 1000 lbf accounting for the beam weight (see Figure 4.2). The cross sectional area of the reinforcing bar $A_s$, the width of the concrete beam $b$ and the depth of the concrete beam $h$ form the design variables. The cross sectional area of the reinforcing bar $A_s$ is a discrete variable that must be chosen from the standardized dimensions listed in [1]. The width of the concrete beam $b$ must be an integer variable from a set of [28, 29, 30, 31,..., 38, 39, 40]. The depth of the concrete beam $h$ a is continuous variable, $5 \leq h \leq 10$.

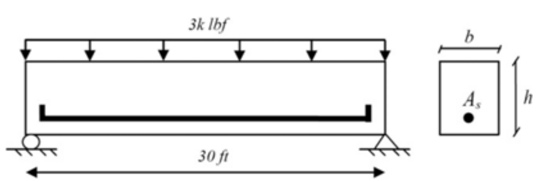


Fig. 4.2: Reinforced concrete beam design structure

**The helical spring design problem** aims to minimize the volume of spring steel wire used to manufacture the spring, shown in Figure 4.3. This problem has three design variables: the number of spring coils, $N$, the outside diameter of the spring, $D$, and the spring wire diameter, $d$. The number of spring coils, $N$, is an integer variable and the outside diameter, $D$, is a continuous variable. The spring wire diameter, $d$, may have only discrete values according to available standard spring steel wire diameters listed in [8].

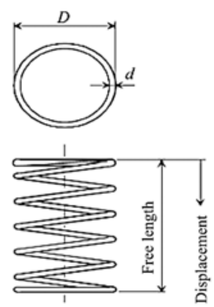


Fig. 4.3: Helical spring design structure

**The stepped cantilever beam design problem** is to minimize the volume of the stepped cantilever beam under the vertical tip force, shown in Figure 4.4. The height and width of the rectangular cross section of each step form the design variables. The problem consists of ten design variables. The width $b_1$ and height $h_1$ of the first step are integer values; the widths $b_2$, $b_3$ of the second and third steps are discrete values, chosen from a set of [2.4, 2.6, 2.8, 3.1]; the heights $h_2$, $h_3$ of the second and third steps are discrete values, chosen from a set of [45.0, 50.0, 55.0, 60.0]. The remaining of the design variables are continuous.

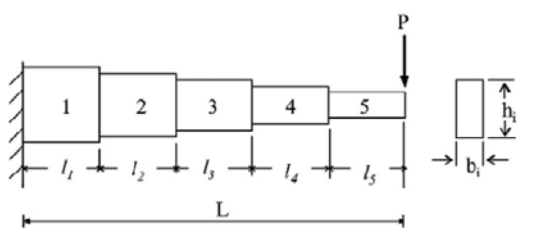


Fig. 4.4: Stepped cantilever beam design structure

The mathematical models of these problems are given in [8].

## 5. Experimental analysis

The proposed E-FA has been implemented in Java programming language and run on a PC with Intel(R) Core(TM) i7-3770K 4.2GHz processor with 16GB of RAM and Windows 8 x 64 Pro operating system. The optimization results obtained by the E-FA were compared with the same of the original FA [8]. The results of the algorithms used for comparison with the proposed E-FA were taken from [8].

### 5.1. Parameter settings

The parameter values adopted by the FA [8] are the following: the size of population *SP* is 25, the value of parameter $\gamma$ is $1/L$, where $L$ is the typical length of design variables, the initial value of attractiveness $\beta_0$ is 1, the initial value of parameter $\alpha$ is 0.5, i.e. the reduction scheme described by Eq.2.7 was followed by reducing the value of parameter $\alpha$ from 0.5 to 0.01. The maximum cycle number *MCN* of 2000 was used for welded beam and stepped cantilever beam problems. For problem reinforced concrete beam *MCN* of 1000 was used, while for problem helical spring *MCN* of 3000 was utilized. The statistical results of the FA were provided over 100 runs.

The parameter values utilized by the proposed E-FA in all the experiments are the following: the size of population *SP* is 25, the value of parameter $\gamma$ is 1, the value of parameter $\beta$ is 1.5 and the initial value of of parameter $\alpha$ is 0.9. For problem welded beam *MCN* of 200 was used, while for problems reinforced concrete beam and helical spring *MCN* of 350 was set. The *MCN* was limited to 2000 for problem stepped cantilever beam. It should be mentioned that although the value 1 for parameter $\beta$ was suitable for most of the FA applications, it was empirically determined that the performance of the proposed E-FA for these problems is sensitive to this parameter value and slightly higher value was more suitable. Each of the experiments was repeated for 100 runs.

### 5.2. Results and discussion

The comparison and discussion of the results are based on obtained better best results and better mean and standard deviation values, as well as lower numbers of evaluations used. The best results demonstrate the ability of an algorithm to reach the optimal result, while mean and standard deviation values indicate the robustness of the algorithm. The total number of evaluations can be considered a measure of computational cost or a convergence rate.

The results for the E-FA are presented using high-precision numbers to allow a correct manual calculation of the objective function values obtained in the experiments. Table 5.1 presents the statistical results obtained by the FA [8] and proposed E-FA. Best results are in bold. When both algorithms had equal results, none was emphasized.

For welded beam problem, the proposed E-FA was able to find the best reported rounded value to reach [14] in each run. On the other hand, the original FA obtained notably worse best solution, as well as the mean result and standard deviation value. In addition, the E-FA reduced the number of evaluations by a factor of 10 in comparison with the same of the FA.

Table 5.1: Comparative results obtained by the FA [8] and proposed E-FA over 100 runs for four structural optimization problems (best results bold)

| Problem | Stats | FA [8] | E-FA |
|---|---|---|---|
| Welded beam | Best | 1.7312065 | **1.7248523** |
| | Mean | 1.8786560 | **1.7248523** |
| | Worst | 2.3455793 | **1.7248523** |
| | St. Dev | 0.2677989 | **6.17E-9** |
| | Eval. | 50000 | **5000** |
| Reinforced concrete beam | Best | 359.2080 | 359.2080 |
| | Mean | 460.706 | **359.2080** |
| | Worst | 669.150 | **359.2080** |
| | St. Dev | 80.73870 | **1.89E-10** |
| | Eval. | 25000 | **8750** |
| Helical spring | Best | 2.658575665 | **2.658559166** |
| | Mean | 4.3835958 | **2.6585592** |
| | Worst | 7.8162919 | **2.6585592** |
| | St. Dev | 4.6076313 | **4.42E-10** |
| | Eval. | 75000 | **8750** |
| Stepped cantilever beam | Best | **63893.52578** | 64578.194035 |
| | Mean | **64144.75312** | 64578.194024 |
| | Worst | **64262.99420** | 64578.194053 |
| | St. Dev | 175.91879 | **5.76E-6** |
| | Eval. | 50000 | 50000 |

For reinforced concrete beam and helical spring problems, the E-FA found the same or slightly better best result and considerably better mean and standard deviation values than the same of FA. In terms of convergence speed, the E-FA performs about three times faster for reinforced concrete beam design problem and about eight times faster than the FA for helical spring design problem.

For stepped cantilever beam problem, the E-FA achieved similar feasible solution of 64578.194 in each run. For this problem, the original FA obtained better best and mean results and worse standard deviation value than the same of E-FA for the same number of objective function evaluations. In addition, in [8] the results are not presented using high-precision numbers to allow a correct manual calculation of the objective function values and also the values of constraints for tested problems. On the other hand, in [8] the values of constraints are not reported for stepped cantilever beam problem. Therefore, it is not clear does the best solution obtained using the FA violates some of the constraints.

Tables 5.2, 5.4, 5.3 and 5.5 present the solution vectors for the best solutions reached by the tested algorithms and the values of the constraints for each tested problem.

The summary results show that the proposed E-FA is able to find the best solutions which are equal or very close to the best found solution reported in the literature for these four structural optimization problems. The E-FA performs

Table 5.2: Parameter and constraint values of the best solutions obtained by the FA [8] and E-FA for welded beam problem (NA means not available)

| | FA [8] | E-FA |
|---|---|---|
| $x_1$ | 0.2015 | 0.20572963705932407 |
| $x_2$ | 3.562 | 3.4704887163500215 |
| $x_3$ | 9.0414 | 9.036623941581052 |
| $x_4$ | 0.2057 | 0.20572963977307454 |
| $g_1(x)$ | NA | -1.26E-5 |
| $g_2(x)$ | NA | -2.05E-4 |
| $g_3(x)$ | NA | -2.71E-9 |
| $g_4(x)$ | NA | -3.432984 |
| $g_5(x)$ | NA | -0.080730 |
| $g_6(x)$ | NA | -0.235540 |
| $g_7(x)$ | NA | -1.24E-05 |
| $f(x)$ | 1.7312 | 1.7248523165044531 |

Table 5.3: Parameter and constraint values of the best solutions obtained by the FA [8] and E-FA for the design of reinforced concrete beam

| | FA [8] | E-FA |
|---|---|---|
| $x_1$ | 6.32 | 6.32 |
| $x_2$ | 34.0 | 34.0 |
| $x_3$ | 8.5000 | 8.500000000000028 |
| $g_1(x)$ | -0.2241 | -0.2241 |
| $g_2(x)$ | 0 | -1.33E-14 |
| $f(x)$ | 359.2080 | 359.20800000000054 |

better than the FA with respect to the quality and robustness of the results with considerably improved convergence speed for the majority of tested problems.

Additionally, in [8] it was concluded that the original FA is more efficient than other metaheuristic algorithms such as PSO, GA, SA and harmony search (HS). Therefore this conclusion also holds for the proposed E-FA.

Table 5.4: Parameter and constraint values of the best solutions obtained by the FA [8] and E-FA for helical compression spring

| | FA [8] | E-FA |
|---|---|---|
| $x_1$ | 0.283 | 0.283 |
| $x_2$ | 1.223049 | 1.2230410099640818 |
| $x_3$ | 9 | 9 |
| $g_1(x)$ | -1008.02 | -1008.811 |
| $g_2(x)$ | -8.946 | -8.946 |
| $g_3(x)$ | -0.083 | -0.083 |
| $g_4(x)$ | -1.777 | -1.494 |
| $g_5(x)$ | -1.322 | -1.322 |
| $g_6(x)$ | -5.464 | -5.464 |
| $g_7(x)$ | 0 | 0 |
| $g_8(x)$ | 0.0000 | -8.41E-13 |
| $f(x)$ | 2.658575665 | 2.6585591659701966 |

Table 5.5: Parameter and constraint values of the best solutions obtained by the FA [8] and E-FA for the design of stepped cantilever beam (NA means not available)

| | FA [8] | E-FA |
|---|---|---|
| $x_1$ | 3 | 3 |
| $x_2$ | 60 | 60 |
| $x_3$ | 3.1 | 3.1 |
| $x_4$ | 55 | 55 |
| $x_5$ | 2.6 | 2.6 |
| $x_6$ | 50 | 50 |
| $x_7$ | 2.205 | 2.2808874178752334 |
| $x_8$ | 44.091 | 45.61774832356247 |
| $x_9$ | 1.750 | 1.7497570126997908 |
| $x_{10}$ | 34.995 | 34.99514024843488 |
| $g_1(x)$ | NA | -1.38E-05 |
| $g_2(x)$ | NA | -1359.050 |
| $g_3(x)$ | NA | -153.846 |
| $g_4(x)$ | NA | -1203.412 |
| $g_5(x)$ | NA | -111.111 |
| $g_6(x)$ | NA | -1.16E-10 |
| $g_7(x)$ | NA | 0.0 |
| $g_8(x)$ | NA | -2.258 |
| $g_9(x)$ | NA | -0.769 |
| $g_{10}(x)$ | NA | -1.49E-08 |
| $g_{11}(x)$ | NA | -3.18E-09 |
| $f(x)$ | 63893.52 | 64578.19402431244 |

## 6. Conclusion

The firefly algorithm (FA) is a prominent new swarm intelligence metaheuristic which has been successfully used to solve a huge number of hard optimization problems. In this paper an enhanced firefly algorithm (E-FA) is proposed in order to improve the performance of the FA for mixed variable structural optimization problems. The E-FA employs Deb's rules in order to handle the constraint instead of penalty approach which is used in the original FA. Also, in order to improve the quality of the solutions and convergence speed, the E-FA uses the geometric progression reduction scheme similar to the cooling schedule of simulated annealing.

The proposed approach was then tested on four structural optimization problems. The comparison results show that the proposed E-FA, for the majority of tested problems, found similar or better best solutions, with lower mean and standard deviations, and with significantly faster convergence. From this research, it can be concluded that the results obtained by the proposed E-FA are very promising and encourage further research for applying it to some other constrained optimization problems, as well as its extension to solve the multi-objective problems.

## REFERENCES

1. A. H. Amir and T. Hasegawa: *Nonlinear mixed-discrete structural optimization*. Journal of Structural Engineering **115** (3) (1989), 626-646.
2. A. Baykasoğlu, L. Özbakr and P. Tapkan: *Artificial Bee Colony Algorithm and Its Application to Generalized Assignment Problem* . In: Swarm Intelligence: Focus on Ant and Particle Swarm Optimization (F. T. S. Chan, M. K. Tiwari, eds.), I-Tech Education and Publishing, Vienna, Austria, 2007, pp. 113-144.
3. H. G. Beyer and H. P. Schwefel: *Evolution strategies - A comprehensive introduction*. Natural computing **1** (1) (2002), 3-52.
4. I. Brajevic: *Crossover-based artificial bee colony algorithm for constrained optimization problems*. Neural Computing and Applications, DOI: 10.1007/s00521-015-1826-y, Springer London, 2015, 1–15.
5. I. Brajevic and M. Tuba: *Cuckoo Search and Firefly Algorithm Applied to Multilevel Image Thresholding*. In: Cuckoo Search and Firefly Algorithm (X.S. Yang, eds.), Studies in Computational Intelligence, Volume 516, Springer International Publishing, 2014, pp. 115–139.
6. K. Deb: *An Efficient Constraint-handling Method for Genetic Algorithms*. Computer Methods in Applied Mechanics and Engineering **186** (2–4) (2000), 311–338.
7. I. Fister Jr., M. Perc, S. M. Kamal and I. Fister: *A review of chaos-based firefly algorithms: Perspectives and research challenges*. Applied Mathematics and Computation **252** (2015), 155-165.
8. A. H. Gandomi, X-S. Yang and A. H. Alavi: *Mixed variable structural optimization using Firefly Algorithm*. Computers and Structures **89** (2011), 2325-2336.

9. A. H. Gandomi, X-S. Yang, A. H. Alavi and S. Talatahari: *Bat algorithm for constrained optimization tasks*. Neural Computing and Applications **22** (6) (2013), 1239–1255.
10. A. H. Gandomi, X-S. Yang and A. H. Alavi: *Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems*. Engineering with Computers **29** (2013), 17-35.
11. J. Holland: *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, 1992.
12. D. Karaboga and B. Akay: *A Modified Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems*. Applied Soft Computing **11** (3) (2011), 3021–3031.
13. J. Kennedy and R. C. Eberhart: *Particle swarm optimization*. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.
14. V. V. D. Melo and G. L. C. Carosio: *Investigating Multi-View Differential Evolution for solving constrained engineering design problems*. Expert Systems with Applications **40** (9) (2013), 3370–3377.
15. E. Mezura-Montes and C. A. Coello Coello: *Constraint-Handling in Nature-Inspired Numerical Optimization: Past, Present and Future*. Swarm and Evolutionary Computation **1** (4) (2011), 173-194.
16. S. Severin and J. Rossmann: *A comparison of different metaheuristic algorithms for optimizing blended PTP movements for industrial robots*. In: Intelligent Robotics and Applications (C. Y. Su, S. Rakheja, H. Liu, eds.), Lecture Notes in Computer Science, Volume 7508, 2012, pp. 321–330.
17. R. Storn and K. Price: *Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces*. Journal of Global Optimization **11** (4) (1997), 341–359.
18. S. Talatahari, A. H. Gandomi and G. J. Yun: *Optimum design of tower structures using firefly algorithm*. The Structural Design of Tall and Special Buildings **23** (5) (2014), 350-361.
19. M. Črepinšek, L. Shih-Hsi and M. Mernik: *Exploration and exploitation in evolutionary algorithms: A survey*. ACM Computing Surveys (CSUR) **45** (3) (2013), 1–33.
20. X-S. Yang: *Review of Metaheuristics and Generalized Evolutionary Walk Algorithm*. Int. J. Bio-Inspired Computation **3**(2) (2011), 77-84.
21. X-S. Yang: *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, United Kingdom, 2010.
22. X-S. Yang: *Firefly algorithms for multimodal optimization*. In: Stochastic Algorithms: Foundations and Applications (O. Watanabe, T. Zeugmann, eds.), Lecture Notes in Computer Science, Volume 5792, Springer Berlin Heidelberg, 2009, pp. 169–178.
23. X-S. Yang: *Firefly algorithm: recent advances and applications*. Int. J. of Swarm Intelligence **1**(1) (2013), 36–50.
24. X-S. Yang: *A new metaheuristic bat-inspired algorithm*. In: Nature inspired cooperative strategies for optimization (NISCO 2010) (J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor, eds.), Studies in Computational Intelligence, Volume 284, Springer Berlin Heidelberg, 2010, pp. 65-74.
25. X-S. Yang and S. Deb: *Cuckoo search via Lévy flights*. In: Proc. of World Congress on Nature & Biologically Inspired Computing, 2009, pp. 210–214.

26. X-S. Yang and S. S. S. Hosseini and A. H. Gandomi: *Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect*. Applied Soft Computing **12** 3 (2011), 1180–1186.

27. M. A. Zaman and M. A. Matin: *Nonuniformly spaced linear antenna array design using firefly algorithm*. International Journal of Microwave Science and Technology **2012**, Article ID 256759, (2012), 8 pages.

28. Y. Zhang and L. Wu: *A novel method for rigid image registration based on firefly algorithm*. International Journal of Research and Reviews in Soft and Intelligent Computing (IJRRSIC) **2**(2) (2012), 141–146.

Ivona Brajevic
Faculty of Mathematics
University of Belgrade
Studentski trg 16, 11000 Beograd, Serbia
ivona.brajevic@googlemail.com

Jelena Ignjatović
Department of Mathematics and Computer Science
Faculty of Sciences and Mathematics
University of Niš
Višegradska 33, 18000 Niš, Serbia
jekaignjatovic73@gmail.com